REMARKS

A. Status of the claims

Claims 26-49 are pending, of which claims 26, 32, 38, and 44 are independent. Applicant has not amended the claims in the present response.

B. Summary of the Final Office Action mailed October 02, 2009

The Final Office Action mailed October 02, 2009 included: (i) a rejection of claims 26-28, 30-34, 36-40, and 42-29 under 35 U.S.C. § 102(e) as allegedly anticipated by U.S. Patent 6,877,005 to Hunter et al. ("Hunter"); and (ii) a rejection of claims 29, 35, and 41 under 35 U.S.C. § 103(a) as allegedly obvious in view of Hunter and U.S. Patent 5,727,051 to Holender et al. ("Holender"). Applicant does not concede that Hunter is prior art with respect to the present application, and Applicant reserves the right to antedate the Hunter reference. Applicant thanks the Examiner for the thorough examination, and Applicant responds to the rejections as follows.

C. The §§ 102(e) and 103(a) rejections based on Hunter are clearly erroneous

The rejections of claims 26-49 based Hunter are clearly erroneous for at least the reason that Hunter fails to disclose, teach, or suggest each of the recited claim elements. As a result, Hunter does not anticipate the claims under 35 U.S.C. § 102 as set forth in MPEP § 2131, and any obviousness rejection based on Hunter alone necessarily lacks the factual underpinnings required to support a finding of *prima facie* obviousness under 35 U.S.C. § 103(a) as set forth in MPEP § 2142. At a minimum, Hunter does not disclose, teach, or suggest at least "searching…for an iterative search result using a subsequent key comprising the subtree index found in a preceding search," as recited in representative independent claim 26.

1. Hunter does not disclose, teach, or suggest "searching…for an iterative search result using a subsequent key comprising the subtree index found in a preceding search"

In rejecting the claims based on Hunter, the Final Office Action erroneously asserts that "it is obvious that each iteration of a search is based on previous masks search results [sic] of the preceding index and a new shortened hash index based on another portion of the address." (Final Office Action, p. 2) But contrary to the assertion in the Final Office Action, Hunter explicitly states that "hash table indices for succeeding [searches] may be generated **without waiting for the results of a previous memory access**." (Hunter, col. 4, lines 3-11 (emphasis added)) Therefore, it is clear that each hash index used for each iteration of Hunter's search is not based on the results of a previous search as asserted in the Final Office Action.

The aspect of Hunter cited as anticipating the pending claims is directed to generating the string of bits that Hunter uses for searching a lookup table, i.e., Hunter's "hash index." Contrary to the assertions in the Final Office Action, Hunter does not disclose "searching…for an iterative search result using a subsequent key comprising the subtree index found in a preceding search," as recited in representative claim 26 because Hunter's subsequent searches are not based on the search results of Hunter's previous searches. Instead of "searching…for an iterative search result using a subsequent key comprising the subtree index found in a preceding search," Hunter performs a subsequent search using a hash index created by the application of a hash function to the bit string that results from the application a decimated mask to the lookup address. In particular, Hunter "produc[es] a hash index," i.e., the string of bits that Hunter searches for in the lookup table, "based on the address and a mask" by applying the mask to the lookup address and then applying the hash function to the masked address to create the hash index. (Hunter, col. 7, lines 55-62) "The initial hash index is typically based upon the whole address." (Hunter, col. 7, lines 56-60) After creating the initial hash index, Hunter searches the lookup table for the initial hash index. If the initial hash index is found in the lookup table, then the search function is complete. But if the initial hash index is not found in the lookup table, then Hunter (i) decimates (i.e., shortens) the mask, (ii) applies the decimated (shorter) mask to the lookup address to create a new masked address, (iii) applies the hash function to the new masked address to create a subsequent hash index, and (iv) searches the lookup table for the subsequent hash index. (Hunter, col. 8, lines 5-6; col. 8, line 40 - col. 9, line 13; col. 9, line 14 - col. 10, line 6) If the subsequent hash index is found in the lookup table, then the search function is complete. But if the subsequent hash index is not found in the lookup table, then Hunter performs the mask decimation (to create an even shorter mask), mask application, and hash application steps again to create a new subsequent hash index, and then Hunter searches the lookup table for the new subsequent hash index. "[A]pplying progressively shorter masks" to the lookup address for each successive search causes each of Hunter's subsequent hash indices to be based on a progressively longer portion of the lookup address. (Hunter, col. 7, line 56 - col. 8, line 11) In other words, as the mask gets shorter and shorter for each successive search, the hash index used in each successive search is based on more and more bits of the lookup address. "Because [Hunter's] subsequent masks [are] based upon the address itself" instead of a search result found in a preceding search, Hunter's "hash table indices for succeeding [searches] may be generated **without waiting for the results of a previous [search]**," (Hunter, col. 4, lines 3-6 (emphasis added)) And because Hunter's approach to mask decimation, hash index creation, and hash index searching is "**not dependent upon results of a [search]**" (Hunter, col. 8, lines 8-11 (emphasis added)), Applicant submits that Hunter does not disclose, teach, or suggest "searching…for an iterative search result using a subsequent key comprising the subtree index found in a preceding search," as recited in representative claim 26.

2. Claims 26-49 are allowable over Hunter

Because Hunter does not disclose, expressly or inherently, "searching…for an iterative search result using a subsequent key comprising the subtree index found in a preceding search," as recited in claim 26, Applicant submits that Hunter neither anticipates nor renders obvious claim 26, and that claim 26 is therefore allowable over Hunter. And because claims 32, 38, and 44 recite elements similar to those recited in claim 26, Applicant further submits that claims 32, 38, and 44 are allowable over Hunter for at least the reasons articulated above with respect to claim 26. Additionally, Applicant further submits that claims 27-31, 33-37, and 39-43, and 45-49 are allowable for at least the reason that they depend from allowable claims 26, 32, 38, and 44, respectively.

D. The § 103(a) rejection based on Hunter and Holender is clearly erroneous

Dependent claims 29, 35, and 41 stand rejected under 35 U.S.C. § 103(a) as allegedly obvious over the combination of Hunter and Holender. (Final Office Action, p. 6) Without conceding the merits of the other assertions set forth in the Office Action, the Applicant submits that claims 29, 35, and 41 are allowable over the combination of Hunter and Holender for at least the reason that Hunter does not disclose the elements of independent claims 26, 32, and 38 from which claims 29, 35, and 41 depend, and that the addition of Holender does not overcome the deficiencies of Hunter set forth in Section C, *supra*.

E. Conclusion

Applicant submits that the present application is in condition for allowance. If the Examiner feels that further dialog would advance the application to issuance, the Examiner is invited to telephone the undersigned at (312) 913-0001.

Respectfully submitted,
McDonnell Boehnen Hulbert & Berghoff LLP

Date: December 02, 2009 By: /Jeffrey P. Armstrong /
Jeffrey P. Armstrong
Reg. No. 54,967